

REMARKS

Reconsideration and allowance of the above-identified application are respectfully requested.

Claims 1, 4-14, 17-27, 30-40 and 43-52 are pending in the application.

Claim 4 has been amended to be dependent upon claim 1. Claims 1, 5, 7, 10-12, 14, 18, 20, 23-25, 27, 31, 33, 36-38, 40, 44, 46, and 49-51 have been amended to further clarify that the XML tag from the received message is parsed and interpreted by the router.

Applicant's acknowledges with appreciation the withdrawal of the previous rejection under 35 U.S.C. § 102 over Abjanic.

The rejection of claims 1, 4-14, 17-27, 30-40 and 43-52 under 35 U.S.C. § 103(a) as being unpatentable over Abjanic as applied to claims 1, 14, 27 and 40, in view of U.S. Patent Pub No. 2002/0198974 (Shafer) is respectfully traversed. Applicant respectfully submits that the subject matter of claims 1, 4-14, 17-27, 30-40 and 43-52 is not obvious over the theoretical combination of Abjanic and Shafer for the following reasons.

Each of the independent claims specifies a router with the ability to parse an XML tag from a received message, interpret the parsed XML tag, and route the received message based on the XML instructions. In this regard, the improved router can dynamically access a vocabulary to interpret the parsed XML tag. See page 8, line 20 through page 9, line 9 of the present specification, which teaches that "the vocabulary library 30 is configured for storing vocabulary modules 32 that enable the application resource 28 to interpret the prescribed attributes specified by the XML tags within the message 14. If an XML header 14a includes tags that not recognized by the application resource 28, the application resource 28 may issue a request to an external source, for example the services registry 20, for retrieval of the appropriate vocabulary module 32 for interpretation of the XML tags." (Emphasis added.)

Thus, the presently claimed router can dynamically reference the XML vocabulary library to learn the attribute function and thereby execute an XML tag for a heretofore unrecognized XML based protocol or service (e.g., SOAP, ebXML, Rosetta, etc., See page 6, line 6 of the

present specification.), in order to route the received message to the destination node.

The Examiner admits that Abjanic does not disclose the use of the claimed vocabulary library. The Examiner argues on page 8 of the pending Office Action that “Shafer mentions a render library that contains style sheets, object definition language (ODL) files necessary to render XML output (Shafer Para 0050 and 0053). It would have been obvious to one of ordinary skill in the art to apply Shafer to Abjanic, providing Abjanic the benefit of having a render language library for the client applications to access for rendering support and to render outputs based on the contents of XML style sheets and ODL files as taught by Shafer Paragraph 0053.”

However, Shafer does not teach or suggest retrieving syntax and semantics information from a vocabulary library (render library) for routing the received message to the destination node. Shafer in fact teaches that the render library is in the client, not in the router. See Fig. 5 of Shafer which clearly shows the render library 70 is in the CLI client 46, not in the routing engine 14. See also Fig. 6 of Shafer, which clearly teaches that the XML reply 96 is parsed, displayed 98, and render library accessed 100 in the Client Application, not in the router.

Thus, Shafer teaches that a network administrator can use the render library 70 to configure the router. The router in Shafer does not utilize the render library 70 during routing to understand routing attributes in XML language. Shafer specifically teaches that the render library 70 “renders the extracted information to a human-readable format...” (See paragraph 50 of Shafer).

In contrast, each of the independent claims specify that the router accesses the XML library to interpret the parsed XML tag from the received message to determine where the message is routed. Shafer only teaches that the client device 46 uses the render library 70 to in the client for displaying information, not routing. There is no structure provided in Shafer such that the router can dynamically interpret a parsed XML tag from the received message and determine routing information therefrom.

Paragraphs 50 and 53 of Shafer state the following:

[0050] FIG. 5 is a block diagram depicting a network router management interface in communication with a **CLI client 46** for purposes of illustration. In general, the network router management interface includes software modules

residing on router 10 for interaction with client applications such as CLI client 46. As shown in FIG. 5, **CLI client 56** may include a render engine 68, render library 70, and a display device 72 for displaying rendered output. Render engine 68 parses the XML replies received from management server module 32, and renders the extracted information to a **human-readable format** using render library 70. Render library 70 may contain style sheets, object definition language (ODL) files, and the like, necessary to textually or graphically render the XML output on display device 72. In some instances, the output may simply be rendered as text at the command line, archived to a file, printed, or presented on another type of output medium. In addition, **CLI client 46** receives user input 78 and directs user commands 76 to routing engine 14. User input 78 may take the form of keystrokes, mouse clicks and the like, as entered by a human user associated with a machine executing CLI client application 46.

[0053] In each case, management server module 32 transmits the XML-encoded reply to the **client application**. Management server module 32 first determines whether it is in a default mode or the 'display xml' mode (90). In the default mode, management server module 32 presents output for presentation by the client application in a rendered format. If the display xml mode applies, management server module 32 adds to the XML reply a command (92) that the **client application** should not render the XML reply. Instead, the **client application** should respond by preparing the XML reply for presentation in a raw, unrendered format. With the exception of the "display xml" indication, the XML output emitted by management server module 32 otherwise may be the same in both modes. If the 'display xml' mode does not apply, management server module 32 transmits the XML reply to the client application (94). The **client application** parses the XML reply (96). In the event the XML reply is not accompanied by a 'display xml' mode command (98), the client application accesses the render library (100) for rendering support, and renders output based on the contents of the XML reply (102) and any style sheets, ODL files, or other information provided by the render library. If the XML reply instructs the client application to present the XML reply in an unrendered format, the client application merely presents the XML code without additional textual or graphical rendering (104). [(Emphasis added.) Shafer at paragraphs 50 and 53.]

The Examiner has provided no support for the mere conclusion that the router of Abjanic would use the render library to "for rendering support and to render outputs." From reading paragraphs 50 and 53 above, it is quite clear that Shafer only teaches how a network administrator (**client application**) can use the render library 70 to configure the router. The router in Shafer does not dynamically use the render library 70. There is simply no disclosure in

paragraphs 50 and 53 as to how the router would use a render library to interpret a parsed XML tag from a received message and rout the information based on the XML instructions. Applicant submits that the Examiner is unfairly relying upon the present disclosure to provide this teaching.

Thus, assuming the references were combined, the resulting hypothetical combination rendering to a human-readable format on a screen still would not teach or even suggest the claimed vocabulary library such that the claimed “application resource” can interpret, for example, XML tags, to perform routing functions based on retrieval of syntax and semantics information from a selected vocabulary module. (See also page 8, lines 19-24 of the present specification). Shafer clearly teaches using ODL to display router configured information to users that configured the router. (See Abstract and paragraphs 50 and 53 of Shafer). There is no teaching or even a suggestion in either reference for the router to dynamically use a vocabulary library to interpret routing attributes in XML language.

Hence, the hypothetical combination does not accommodate different XML-based routing protocols as they are developed, but simply assumes a static configuration providing no flexibility in interpreting different XML-based routing protocols.

Further, the client application of Shafer referred to by the Examiner is very different from the claimed application resource that initiates selected application operations for routing the received message. The claimed application resource utilizes the library to interpret XML attributes during routing. The client application of Shafer only “accesses the render library (100) for rendering support” to provide human readable format for a network administrator during configuration of the router. There is no equivalence between the “client application” of Shafer and the presently claimed “application resource.”

Applicant once again submits that the Examiner has improperly used the present disclosure as a template to piece together unrelated elements from the cited references and to use hindsight gleaned from the present application to fill in the gaps of the cited references. “It is impermissible to use the claimed invention as an instruction manual or ‘template’ to piece together the teachings of the prior art so that the claimed invention is rendered obvious.” In re Fritch, 23 USPQ2d 1780, 1784 (Fed. Cir. 1992).

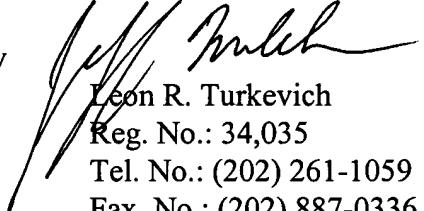
In view of the many differences between the combination of references and the present invention, withdrawal of the Section 103 rejection is respectfully requested.

Since all of the objections and objections of record have been addressed, it is believed that the application is in condition for allowance and Notice to that effect is respectfully requested.

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a), to Deposit Account No. 50-1130, under Order No. 95-466, and please credit any excess fees to such deposit account.

Respectfully submitted,

By


Leon R. Turkevich
Reg. No.: 34,035
Tel. No.: (202) 261-1059
Fax. No.: (202) 887-0336

Customer No. 23164
August 16, 2005